Redes Neurais Artificiais, prof. Hitalo Nascimento

Atividade: Backpropagation

Larissa Vitória Vasconcelos Sousa

Matrícula: 519221

```
import math

x1 = 0.1
x2 = 0.5
w1 = 0.1
w2 = 0.2
w3 = 0.3
w4 = 0.4
w5 = 0.5
w6 = 0.6
w7 = 0.7
w8 = 0.8
b1 = 0.25
b2 = 0.35
n = 0.6
target1 = 0.05
target2 = 0.95

sum_h1 = (0.1 * 0.1) + (0.5 * 0.3) + 0.25
print("\nSoma 1: ", sum_h1)

sum_h2 = (0.1 * 0.2)+(0.5 * 0.4) + 0.25
print("\nSoma 2: ", sum_h2)

output_h1 = 1/(1 + math.exp(-sum_h1))
print("\nOutput H1: ", output_h1)

output_h2 = 1/(1 + math.exp(-sum_h2))
print("\nOutput H2: ", output_h2)

sum_output_1 = (output_h1 * w5) + (output_h2 * w6) + b2
print("\nSoma do output 1: ", sum_output_1)

sum_output_2 = (output_h1 * w7) + (output_h2 * w8) + b2
print("\nSoma do output 2: ", sum_output_2)

output_ho1 = 1/(1 + math.exp(-sum_output_1))
print("\nSoma do output Ho 1: ", output_ho1)

output_ho2 = 1/(1 + math.exp(-sum_output_2))
print("\nSoma do output Ho 2: ", output_ho2)

E1 = 1/2 * (target1 - output_ho1) ** 2
print("\nE1: ", E1)

E2 = 1/2 * (target2 - output_ho2) ** 2
print("\nE2: ", E2)

ErroTotal = E1 + E2
print("\nErro total: ", ErroTotal)

derivadaErroW5 = (output_ho1 - target1) * (output_ho1 * (1 - output_ho1) * output_h1)
print("\nDerivada do erro w5: ", derivadaErroW5)

novo_w5 = w5 - n * derivadaErroW5
print("\nNovo w5: ", novo_w5)

novo_w2 = w2 - n * 0.0001
print("\nNovo w2: ", novo_w2)

novo_w3 = w3-n * 0.00327
print("\nNovo w3: ", novo_w3)

novo_w4 = w4 - n * 0.00672
print("\nNovo w4: ", novo_w4)
```

```
Soma 1:  0.41000000000000003
```

Soma 2: 0.47000000000000003

Output H1: 0.6010878788483698

Output H2: 0.6153837563911821

Soma do output 1: 1.0197741932588942

Soma do output 2: 1.2630685203068044

Soma do output Ho 1: 0.7349286127170142

Soma do output Ho 2: 0.7795538841677558

E1: 0.23456360225922676

E2: 0.014525939201149389

Erro total: 0.24908954146037615

Derivada do erro w5: 0.08020312429316064

Novo w5: 0.4518781254241036

Novo w2: 0.19994

Novo w3: 0.29803799999999997

Novo w4: 0.39596800000000004